```
MMM         MMM     000000000     NNN         NNN  TTTTTTTTTTTTTTT     000000000     RRRRRRRRRRRR
MMM         MMM     000000000     NNN         NNN  TTTTTTTTTTTTTTT     000000000     RRRRRRRRRRRR
MMM         MMM     000000000     NNN         NNN  TTTTTTTTTTTTTTT     000000000     RRRRRRRRRRRR
MMMMMM   MMMMMM  000         000  NNN         NNN        TTT        000         000  RRR         RRR
MMMMMM   MMMMMM  000         000  NNN         NNN        TTT        000         000  RRR         RRR
MMMMMM   MMMMMM  000         000  NNN         NNN        TTT        000         000  RRR         RRR
MMM   MMM   MMM  000         000  NNNNNN      NNN        TTT        000         000  RRR         RRR
MMM   MMM   MMM  000         000  NNNNNN      NNN        TTT        000         000  RRR         RRR
MMM   MMM   MMM  000         000  NNNNNN      NNN        TTT        000         000  RRR         RRR
MMM         MMM  000         000  NNN   NNN   NNN        TTT        000         000  RRRRRRRRRRRR
MMM         MMM  000         000  NNN   NNN   NNN        TTT        000         000  RRRRRRRRRRRR
MMM         MMM  000         000  NNN   NNN   NNN        TTT        000         000  RRRRRRRRRRRR
MMM         MMM  000         000  NNN      NNNNNN        TTT        000         000  RRR   RRR
MMM         MMM  000         000  NNN      NNNNNN        TTT        000         000  RRR   RRR
MMM         MMM  000         000  NNN      NNNNNN        TTT        000         000  RRR   RRR
MMM         MMM  000         000  NNN         NNN        TTT        000         000  RRR      RRR
MMM         MMM  000         000  NNN         NNN        TTT        000         000  RRR      RRR
MMM         MMM  000         000  NNN         NNN        TTT        000         000  RRR      RRR
MMM         MMM     000000000     NNN         NNN        TTT           000000000     RRR         RRR
MMM         MMM     000000000     NNN         NNN        TTT           000000000     RRR         RRR
MMM         MMM     000000000     NNN         NNN        TTT           000000000     RRR         RRR
```

```
**FILE**ID**SHODEF                                              G 13
```

```
  SSSSSSSS  HH      HH    OOOOOO    DDDDDDDD    EEEEEEEEEE  FFFFFFFFFF
  SSSSSSSS  HH      HH    OOOOOO    DDDDDDDD    EEEEEEEEEE  FFFFFFFFFF
SS          HH      HH  OO      OO  DD      DD  EE          FF
SS          HH      HH  OO      OO  DD      DD  EE          FF
SS          HH      HH  OO      OO  DD      DD  EE          FF
SS          HH      HH  OO      OO  DD      DD  EE          FF
  SSSSSS    HHHHHHHHHH  OO      OO  DD      DD  EEEEEEEE    FFFFFFFF
  SSSSSS    HHHHHHHHHH  OO      OO  DD      DD  EEEEEEEE    FFFFFFFF
        SS  HH      HH  OO      OO  DD      DD  EE          FF
        SS  HH      HH  OO      OO  DD      DD  EE          FF
        SS  HH      HH  OO      OO  DD      DD  EE          FF              ....
        SS  HH      HH  OO      OO  DD      DD  EE          FF              ....
SSSSSSSS    HH      HH    OOOOOO    DDDDDDDD    EEEEEEEEEE  FF              ....
SSSSSSSS    HH      HH    OOOOOO    DDDDDDDD    EEEEEEEEEE  FF              ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
SHO
V04
```

Table of contents

SHO
V04

63

63

63

```
0000          .TITLE  SHODEF  - MONITOR SHOW DEFAULT Command
0000          .IDENT  'V04-000'
0000
0000  ;
0000  ;****************************************************************************
0000  ;*                                                                          *
0000  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000  ;*  ALL RIGHTS RESERVED.                                                    *
0000  ;*                                                                          *
0000  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000  ;*  TRANSFERRED.                                                            *
0000  ;*                                                                          *
0000  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000  ;*  CORPORATION.                                                            *
0000  ;*                                                                          *
0000  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000  ;*                                                                          *
0000  ;*                                                                          *
0000  ;****************************************************************************
0000
0000  ;++
0000  ; FACILITY:  VAX/VMS MONITOR Utility
0000  ;
0000  ; ABSTRACT:
0000  ;
0000  ;       The SHODEF module executes the SHOW DEFAULT subcommand
0000  ;       of the MONITOR utility. It is called by the CLE (Command
0000  ;       Language Editor).
0000  ;
0000  ; ENVIRONMENT:
0000  ;
0000  ;       User mode, IPL 0, unprivileged.
0000  ;
0000  ; AUTHOR: Thomas L. Cafarella, March, 1983
0000  ;
0000  ; MODIFIED BY:
0000  ;
0000  ;       V03-003 PRS1016         Paul R. Senn            04-Apr-1984     14:00
0000  ;               Use $PARSE to expand filespecs and hide passwords.
0000  ;
0000  ;       V03-003 PRS1013         Paul R. Senn            28-Mar-1984     14:00
0000  ;               Give SHOW DEFAULT the ability to handle multiple input files.
0000  ;
0000  ;       V03-002 PRS1011         Paul R. Senn            29-Feb-1984     14:00
0000  ;               add /FLUSH_INTERVAL qualifier
0000  ;
0000  ;       V03-001 PRS1001         Paul R. Senn            27-Dec-1983     16:00
0000  ;               Make default interval = 6 for ALL classes Pseudo-class
0000  ;                       live requests.
0000  ;
```

```
                                        0000    58 ;--
```

```
    0000          .SBTTL  DECLARATIONS
00000000          .PSECT  MONDATA,QUAD,NOEXE
    0000    ;
    0000    ; INCLUDE FILES:
    0000    ;
    0000
    0000          $CDBDEF                              ; Define Class Descriptor Block
    0000          $MRBDEF                              ; Define Monitor Request Block
    0000          $MONDEF                              ; Monitor Recording File Definitions
    0000          $DSCDEF                              ; Descriptor Definitions
    0000          $IFBDEF                              ; File descriptor table definitions
    0000
    0000    ;
    0000    ; MACROS:
    0000    ;
    0000
    0000    ;
    0000    ; Local Macro Definitions
    0000    ;
    0000
    0000    ;
    0000    ; ALLOC Macro - Dynamically allocate space on the stack.
    0000    ;
    0000
    0000          .MACRO  ALLOC   LENGTH,RSLDESC,RSLBUF
    0000          SUBL    #<LENGTH+3>&<^C3>,SP
    0000          .IF     NB,RSLBUF
    0000          MOVL    SP,RSLBUF
    0000          .ENDC
    0000          PUSHL   SP
    0000          PUSHL   #LENGTH
    0000          MOVL    SP,RSLDESC
    0000          .ENDM   ALLOC
    0000
    0000    ;
    0000    ; EQUATED SYMBOLS:
    0000    ;
    0000
```

SHO
V04

```
                                         0000     99 ;
                                         0000    100 ; OWN STORAGE
                                         0000    101 ;
                                         0000    102
61 76 20 64 65 64 72 6F 63 65 72 00'     0000    103 RV_STR:         .ASCIC  \recorded value\     ; Text for playback values
                            65 75 6C     000C
                                  0E     0000
6D 69 74 20 74 6E 65 72 72 75 63 00'     000F    104 CT_STR:         .ASCIC  \current time\       ; Text for /BEGINNING
                                  65     001B
                                  0C     000F
   65 74 69 6E 69 66 65 64 6E 69 00'     001C    105 ID_STR:         .ASCIC  \indefinite\         ; Text for /ENDING
                                  0A     001C
41 21 3C 39 21 2F 0000002F'010E0000'     0027    106 CS_SEG1:        .ASCID  \/!9<!AS!> = !27<!\  ; Fixed segment 1 of FAOL control str
   21 3C 37 32 21 20 3D 20 3E 21 53      0035
32 31 21 2F 3E 21 00000048'010E0000'     0040    107 CS_SEG2:        .ASCID  \!>/!12<!AS!> = !\   ; Fixed segment 2 of FAOL control str
      21 20 3D 20 3E 21 53 41 21 3C      004E
      3C 39 21 2F 00000060'010E0000'     0058    108 CS_SEG3:        .ASCID  \/!9<\               ; Fixed segment 3 of FAOL control str
   3E 21 53 41 21 0000006C'010E0000'     0064    109 CS_SEG4:        .ASCID  \!AS!>\              ; Fixed segment 4 of FAOL control str
53 41 21 20 3D 20 00000079'010E0000'     0071    110 CS_SEG5:        .ASCID  \ = !AS\             ; Fixed segment 5 of FAOL control str
21 3C 34 31 21 2F 00000087'010E0000'     007F    111 CS_SEG6:        .ASCID  \/!14<!AS!> = !\     ; Fixed segment 6 of FAOL control str
               21 20 3D 20 3E 21 53 41   008D
20 20 20 20 20 20 0000009D'010E0000'     0095    112 CS_SEG7:        .ASCID  \               !AS\ ; Fixed segment 7 of FAOL control str
      53 41 21 20 20 20 20 20 20 20      00A3
                                         00AD    113
41 21 3C 36 32 21 000000B5'010E0000'     00AD    114 CL_SEG1:        .ASCID  \!26<!AC!>\          ; Fixed seg 1 of classes FAOL ctrl str
                            3E 21 43     00BB
41 21 3C 36 32 21 000000C6'010E0000'     00BE    115 CL_SEG2:        .ASCID  \!26<!AC/!AS!>\      ; Fixed seg 2 of classes FAOL ctrl str
                3E 21 53 41 21 2F 43     00CC
65 73 73 61 6C 43 000000DB'010E0000'     00D3    116 CLASS_HDG:      .ASCID  \Classes:\           ; Heading line for classes
                               3A 73     00E1
65 73 73 61 6C 43 000000EB'010E0000'     00E3    117 NO_CLASS_HDG:   .ASCID  \Classes: none\      ; Heading line for "no classes"
                65 6E 6F 6E 20 3A 73     00F1
                                         00F8    118
                                         00F8    119 SHOW_FAB:       $FAB, -                      ; FAB for $PARSE to show filespecs
                                         00F8    120                 FOP=NAM,-
                                         00F8    121                 NAM=SHOW_NAM
                                         0148    122 SHOW_NAM:       $NAM, -                      ; NAM for $PARSE
                                         0148    123                 ESA=SHOW_FILESPEC,-
                                         0148    124                 ESS=NAM$C_MAXRSS,-
                                         0148    125                 NOP=SYNCHK                   ; syntax check only (don't open file)
                                         01A8    126
                           000002A7      01A8    127 SHOW_FILESPEC:  .BLKB   NAM$C_MAXRSS         ; space for expanded filespec
                                         02A7    128
                           000000FF      02A7    129 SHOW_SPEC_D:    .LONG   . - SHOW_FILESPEC    ; descriptor for expanded filespec
                           000001A8'     02AB    130                 .LONG   SHOW_FILESPEC
                                         02AF    131
                           00000000      02AF    132 ERROR_QUAL:     .LONG   0                    ; address of qualifier for filespec
                                         02B3    133                                              ; which contains a syntax error.
                                         02B3    134
```

```
    02B3          .SBTTL  SHODEF_CMD - MONITOR SHOW DEFAULT command
00000000          .PSECT  $$MONCODE,NOWRT,EXE
    0000  ;++
    0000  ;
    0000  ; FUNCTIONAL DESCRIPTION:
    0000  ;
    0000  ;       This routine uses the SCRPKG to display lines in response
    0000  ;       to a SHOW subcommand. All qualifiers and their current values
    0000  ;       are shown, as well as all selected classes.
    0000  ;
    0000  ; INPUTS:
    0000  ;
    0000  ;       None
    0000  ;
    0000  ; IMPLICIT INPUTS:
    0000  ;
    0000  ;       SCRDSC - quadword string descriptor for buffer required by SCRPKG.
    0000  ;       CURR_MRBPTR - pointer to the "current" MRB (Monitor Request Block).
    0000  ;       QUALPTR     - pointer to the Qualifier Descriptors block.
    0000  ;       INTERVAL_DEFAULT - default value for /INTERVAL qualifier.
    0000  ;       ALLCL_INT_DEFAULT - default value for /INTERVAL qualifier for ALL class.
    0000  ;       VIEWING_DEFAULT  - default value for /VIEWING_TIME qualifier.
    0000  ;       MAX_CLASS_NO     - highest MONITOR class number.
    0000  ;
    0000  ; OUTPUTS:
    0000  ;
    0000  ;       None
    0000  ;
    0000  ; IMPLICIT OUTPUTS:
    0000  ;
    0000  ;       SHOW command display is sent to the terminal.
    0000  ;
    0000  ; ROUTINE VALUE:
    0000  ;
    0000  ;       R0 = SS$_NORMAL, or called routine error status
    0000  ;
    0000  ; SIDE EFFECTS:
    0000  ;
    0000  ;       none
    0000  ;
    0000  ; REGISTER USAGE:
    0000  ;
    0000  ;       R0,R1,R2,R4,R5 = scratch, used by MOVC3
    0000  ;       R3  = FAOL control string index
    0000  ;       R7  = pointer to MRB (Monitor Request Block)
    0000  ;       R8  = pointer to Qualifier Descriptors
    0000  ;       R9  = FAOL parameter list index
    0000  ;       R10 = address of descriptor for FAOL parameter list
    0000  ;       R11 = address of descriptor for FAOL control string
    0000  ;
    0000  ;--
    0000
```

```
                                       0000
                                 0FFC  0000    .ENTRY  SHODEF_CMD, ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                                       0002
                                       0002
      00000000'EF   00000000'8F    D0  0002            MOVL    #MNR$_SHOWERR,CURR_ERRCODE ; Set up signaled error code
                    000002AF'EF    D4  000D            CLRL    ERROR_QUAL                 ; clear address of bad qualifier
                                       0013    ;
                                       0013    ; Set up a SCRPKG buffered output stream directed to SYS$OUTPUT
                                       0013    ;
                                       0013
                             00    DD  0013            PUSHL   #0                         ; Stack stream identifier
                 00000000'GF  01    FB  0015            CALLS   #1,G^SCR$SET_OUTPUT        ; Establish output stream
                          03 50    E8  001C            BLBS    R0,10$                     ; Branch if status OK
                           03C8    31  001F            BRW     SHD_ERR                    ; Else go exit with error
                                       0022
                                       0022    10$:
                    00000000'EF    7F  0022            PUSHAQ  SCRDSC                     ; Push this routine's buffer addr
                 00000000'GF  01    FB  0028            CALLS   #1,G^LIB$SET_BUFFER        ; Set buffering mode
                          03 50    E8  002F            BLBS    R0,20$                     ; Branch if status OK
                           03B5    31  0032            BRW     SHD_ERR                    ; Else go exit with error
                                       0035
                                       0035    20$:
                                       0035            ALLOC   40,R10,R9                  ; Allocate an FAOL parameter list
                                       0042            ALLOC   80,R11,R3                  ; Allocate an FAOL control string
               57   00000000'EF    D0  0057            MOVL    CURR_MRBPTR,R7             ; Load up ptr to MRB
               58   00000000'EF    D0  005E            MOVL    QUALPTR,R8                 ; ... and ptr to qualifier descriptors
                                       0065
                                       0065    ;
                                       0065    ; Show /BEGINNING qualifier
                                       0065    ;
                                       0065
            83     2F21 8F         B0  0065            MOVW    #^A\!/\,(R3)+              ; Move "new-line" to FAOL control string
63  0000002F'EF   00000027'EF      28  006A            MOVC3   CS_SEG1,CS_SEG1+8,(R3)     ; Move fixed segment into control string
                     89    68      DE  0076            MOVAL   QUAL$L_BEG(R8),(R9)+       ; /BEGINNING qual name to FAOL prmlst
                     50    67      7D  0079            MOVQ    MRB$Q_BEGINNING(R7),R0     ; Test /BEGINNING defaulted ?
                           0A      13  007C            BEQL    30$                        ; Branch if so
            83     4425 8F         B0  007E            MOVW    #^A/%D/,(R3)+              ; FAO date-time directive to ctrstr
                     89    67      DE  0083            MOVAL   MRB$Q_BEGINNING(R7),(R9)+  ; BEGINNING time to FAOL prmlst
                           1A      11  0086            BRB     SHO_INT                    ; Go set up /INTERVAL
                                       0088    30$:                                       ; /BEGINNING defaulted
            83     4341 8F         B0  0088            MOVW    #^A/AC/,(R3)+              ; FAO cstring directive to ctrstr
                        1C A7      D5  008D            TSTL    MRB$A_INPUT(R7)            ; Live or Playback ?
                           09      13  0090            BEQL    40$                        ; Go do live
               89   00000000'EF    DE  0092            MOVAL   RV_STR,(R9)+               ; Playback -- cstring ptr to FAOL prmlst
                           07      11  0099            BRB     SHO_INT                    ; Go set up /INTERVAL
                                       009B    40$:                                       ; Live
               89   0000000F'EF    DE  009B            MOVAL   CT_STR,(R9)+               ; "Current time" cstring ptr to prmlst
```

```
                                        00A2   237 ;
                                        00A2   238 ; Show /INTERVAL qualifier
                                        00A2   239 ;
                                        00A2   240
                                        00A2   241 SHO_INT:
63   00000048'EF     00000040'EF   28   00A2   242          MOVC3   CS_SEG2,CS_SEG2+8,(R3)     ; Move fixed segment into control string
                          89  10 A8   DE   00AE   243          MOVAL   QUAL$L_INT(R8),(R9)+       ; /INTERVAL qual name to FAOL prmlst
                              10 A7   D5   00B2   244          TSTL    MRB$L_INTERVAL(R7)         ; /INTERVAL defaulted ?
                                 0B   13   00B5   245          BEQL    10$                        ; Branch if so
                    83    4C5A 8F   B0   00B7   246          MOVW    #^A/ZL/,(R3)+              ; FAO decimal directive to ctrstr
                          89  10 A7   D0   00BC   247          MOVL    MRB$L_INTERVAL(R7),(R9)+ ; INTERVAL value to FAOL prmlst
                                 2D   11   00C0   248          BRB     30$                        ; Continue.....
                                           00C2   249 10$:                                         ; /INTERVAL defaulted
                              1C A7   D5   00C2   250          TSTL    MRB$A_INPUT(R7)            ; Live or Playback ?
                                 0E   13   00C5   251          BEQL    20$                        ; Go do live
                    83    4341 8F   B0   00C7   252          MOVW    #^A/AC/,(R3)+              ; FAO cstring directive to ctrstr
               89   00000000'EF   DE   00CC   253          MOVAL   RV_STR,(R9)+               ; Playback -- cstring ptr to FAOL prmlst
                                 1A   11   00D3   254          BRB     30$                        ; Continue.....
                                           00D5   255 20$:                                         ; Live
                    83    4C5A 8F   B0   00D5   256          MOVW    #^A/ZL/,(R3)+              ; FAO decimal directive to ctrstr
                  09 43 A7    0A   E0   00DA   257          BBS     #MRB$V_ALL_CLASS,MRB$W_FLAGS(R7),25$ ; Special default for ALL
               89   00000000'8F   D0   00DF   258          MOVL    #INTERVAL_DEFAULT,(R9)+ ; Default INTERVAL value to prmlst
                                 07   11   00E6   259          BRB     30$                        ; Continue.....
                                           00E8   260 25$:
               89   00000000'8F   D0   00E8   261          MOVL    #ALLCL_INT_DEFAULT,(R9)+ ; Default INTERVAL value for ALL class
                                           00EF   262
                                           00EF   263 30$:
                              FC A9   DD   00EF   264          PUSHL   -4(R9)                     ; Save interval val for /VIEWING_TIME
                                           00F2   265
                                           00F2   266 ;
                                           00F2   267 ; Display a line showing /BEGINNING and /INTERVAL
                                           00F2   268 ;
                                           00F2   269
                               03FB   30   00F2   270          BSBW    SHOW_SINGLE                ; Show the line, single-spaced
                              03 50   E8   00F5   271          BLBS    R0,SHO_END                 ; Go on to /ENDING if status OK
                               02EF   31   00F8   272          BRW     SHD_ERR                    ; Otherwise, go exit
```

```
                                       00FB    ;
                                       00FB    ; Show /ENDING qualifier
                                       00FB    ;
                                       00FB
                                       00FB    SHO_END:
                  53   04 AB    D0     00FB            MOVL    4(R11),R3                ; Point to beginning of FAOL ctrl string
                  59   04 AA    D0     00FF            MOVL    4(R10),R9                ; Point to beginning of FAOL parm list
                                       0103
63   0000002F'EF   00000027'EF   28     0103            MOVC3   CS_SEG1,CS_SEG1+8,(R3)   ; Move fixed segment into control string
                  89   08 A8    DE     010F            MOVAL   QUAL$L_END(R8),(R9)+     ; /ENDING qual name to FAOL prmlst
                  50   08 A7    7D     0113            MOVQ    MRB$Q_ENDING(R7),R0      ; Test /ENDING defaulted ?
                          0B    13     0117            BEQL    10$                      ; Branch if so
                83   4425 8F    B0     0119            MOVW    #^A/%D/,(R3)+            ; FAO date-time directive to ctrstr
                  89   08 A7    DE     011E            MOVAL   MRB$Q_ENDING(R7),(R9)+   ; ENDING time to FAOL prmlst
                          1A    11     0122            BRB     SHO_VIEW                 ; Go set up /VIEWING_TIME
                                       0124    10$:                                     ; /ENDING defaulted
                83   4341 8F    B0     0124            MOVW    #^A/AC/,(R3)+            ; FAO cstring directive to ctrstr
                       1C A7    D5     0129            TSTL    MRB$A_INPUT(R7)          ; Live or Playback ?
                          09    13     012C            BEQL    20$                      ; Go do live
           89   00000000'EF     DE     012E            MOVAL   RV_STR,(R9)+             ; Playback -- cstring ptr to FAOL prmlst
                          07    11     0135            BRB     SHO_VIEW                 ; Go set up /VIEWING_TIME
                                       0137    20$:                                     ; Live
           89   0000001C'EF     DE     0137            MOVAL   ID_STR,(R9)+             ; "Indefinite" cstring ptr to prmlst
                                       013E
                                       013E    ;
                                       013E    ; Show /VIEWING_TIME qualifier
                                       013E    ;
                                       013E
                                       013E    SHO_VIEW:
63   00000048'EF   00000040'EF   28     013E            MOVC3   CS_SEG2,CS_SEG2+8,(R3)   ; Move fixed segment into control string
                  89   20 A8    DE     014A            MOVAL   QUAL$L_VIEW(R8),(R9)+    ; Qualifier name to FAOL prmlst
                83   4C5A 8F    B0     014E            MOVW    #^A/ZL7,(R3)+            ; FAO decimal directive to ctrstr
                       18 A7    D5     0153            TSTL    MRB$L_VIEWING_TIME(R7)   ; /VIEWING_TIME defaulted ?
                          06    13     0156            BEQL    10$                      ; Branch if so
                  89   18 A7    D0     0158            MOVL    MRB$L_VIEWING_TIME(R7),(R9)+ ; VIEWING_TIME value to FAOL prmlst
                          11    11     015C            BRB     30$                      ; Continue.....
                                       015E    10$:                                     ; /VIEWING_TIME defaulted
                       1C A7    D5     015E            TSTL    MRB$A_INPUT(R7)          ; Live or Playback ?
                          09    13     0161            BEQL    20$                      ; Go do live
           89   00000000'8F     D0     0163            MOVL    #VIEWING_DEFAULT,(R9)+   ; Default VIEWING_TIME value to prmlst
                          03    11     016A            BRB     30$                      ; Continue.....
                                       016C    20$:                                     ; Live
                     89   8E    D0     016C            MOVL    (SP)+,(R9)+              ; Pop saved /INTERVAL value to prmlst
                                       016F    30$:
                                       016F
                                       016F    ;
                                       016F    ; Display a line showing /ENDING and /VIEWING_TIME
                                       016F    ;
                                       016F
                        0382    30     016F            BSBW    SHOW_DOUBLE              ; Show the line, double-spaced
                       03 50    E8     0172            BLBS    R0,SHO_FLUSH             ; Go on to file qualifiers if status OK
                        0272    31     0175            BRW     SHO_ERR                  ; Otherwise, go exit
                                       0178
```

```
                                     0178      ;
                                     0178      ; Show /FLUSH_INTERVAL qualifier
                                     0178      ;
                                     0178
                                     0178      SHO_FLUSH:
                  53  04 AB    D0    0178              MOVL    4(R11),R3                   ; Point to beginning of FAOL ctrl string
                  59  04 AA    D0    017C              MOVL    4(R10),R9                   ; Point to beginning of FAOL parm list
                                     0180
63    00000087'EF    0000007F'EF  28 0180              MOVC3   CS_SEG6,CS_SEG6+8,(R3)      ; Move fixed segment into control string
                  89  18 A8    DE    018C              MOVAL   QUAL$L_FLUSH(R8),(R9)+      ; Qualifier name to FAOL prmlst
               83   4C5A 8F    B0    0190              MOVW    #^A/ZL7,(R3)+               ; FAO decimal directive to ctrstr
                      14 A7    D5    0195              TSTL    MRB$L_FLUSH(R7)             ; /FLUSH defaulted ?
                         06    13    0198              BEQL    10$                         ; Branch if so
                  89  14 A7    D0    019A              MOVL    MRB$L_FLUSH(R7),(R9)+       ; FLUSH value to FAOL prmlst
                         07    11    019E              BRB     30$                         ; Continue.....
                                     01A0      10$:                                        ; /FLUSH defaulted
             89   00000000'8F  D0    01A0              MOVL    #FLUSH_INT_DEFAULT,(R9)+    ; Default FLUSH value to prmlst
                                     01A7      30$:
                                     01A7
                                     01A7      ;
                                     01A7      ; Display a line showing /FLUSH_INTERVAL
                                     01A7      ;
                                     01A7
                       034A    30    01A7              BSBW    SHOW_DOUBLE                 ; Show the line, double-spaced
                      03 50    E8    01AA              BLBS    R0,SHO_FILES                ; Go on to file qualifiers if status OK
                       023A    31    01AD              BRW     SHD_ERR                     ; Otherwise, go exit
                                     01B0
                                     01B0      ;
                                     01B0      ; Show qualifiers which always have string values
                                     01B0      ; (if they are present). These are typically qualifiers
                                     01B0      ; with file specs as values.
                                     01B0      ;
                                     01B0
                                     01B0      SHO_FILES:
                                     01B0
                                     01B0              ALLOC   8,R0,R6                     ; Allocate a pair of longwords to pass
                                     01BD                                                  ; ... as input parameter in R6 to
                                     01BD                                                  ; ... SHOW_FILE_QUAL
                                     01BD      ;
                                     01BD      ; At this point, MRB$A_INPUT contains the address of the IFB table, or 0
                                     01BD      ; if there is no current default. (Later, MRB$A_INPUT will be changed
                                     01BD      ; to be the address of a single file spec descriptor, unless we are
                                     01BD      ; doing a multi-file summary.)
                                     01BD      ;
                  66  1C A7    D0    01BD              MOVL    MRB$A_INPUT(R7),(R6)        ; Load addr of qualifier value descr
               04 A6  28 A8    DE    01C1              MOVAL   QUAL$E_INP(R8),4(R6)        ; Load addr of qualifier name descr
                       0045    30    01C6              BSBW    SHOW_INPUT_QUAL             ; display input qualifier
                      3F 50    E9    01C9              BLBC    R0,SF_ERR                   ; Go return if error
                                     01CC
                  66  24 A7    D0    01CC              MOVL    MRB$A_RECORD(R7),(R6)       ; Load addr of qualifier value descr
               04 A6  38 A8    DE    01D0              MOVAL   QUAL$E_REC(R8),4(R6)        ; Load addr of qualifier name descr
                       00D6    30    01D5              BSBW    SHOW_FILE_QUAL              ; Show a line for /RECORD
                      30 50    E9    01D8              BLBC    R0,SF_ERR                   ; Go return if error
                                     01DB
                  66  20 A7    D0    01DB              MOVL    MRB$A_DISPLAY(R7),(R6)      ; Load addr of qualifier value descr
               04 A6  30 A8    DE    01DF              MOVAL   QUAL$E_DISP(R8),4(R6)       ; Load addr of qualifier name descr
                       00C7    30    01E4              BSBW    SHOW_FILE_QUAL              ; Show a line for /DISPLAY
```

```
            21 50   E9  01E7   385          BLBC    R0,SF_ERR                   ; Go return if error
                        01EA   386
      66    28 A7   D0  01EA   387          MOVL    MRB$A_SUMMARY(R7),(R6)      ; Load addr of qualifier value descr
   04 A6    40 A8   DE  01EE   388          MOVAL   QUAL$[_SUMM(R8),4(R6)       ; Load addr of qualifier name descr
             00B8   30  01F3   389          BSBW    SHOW_FILE_QUAL              ; Show a line for /SUMMARY
            12 50   E9  01F6   390          BLBC    R0,SF_ERR                   ; Go return if error
                        01F9   391
      66    2C A7   D0  01F9   392          MOVL    MRB$A_COMMENT(R7),(R6)      ; Load addr of qualifier value descr
   04 A6    48 A8   DE  01FD   393          MOVAL   QUAL$[_COMM(R8),4(R6)       ; Load addr of qualifier name descr
             00FF   30  0202   394          BSBW    SHOW_QUAL                   ; Show a line for /COMMENT
            03 50   E9  0205   395          BLBC    R0,SF_ERR                   ; Branch on error
             013D   31  0208   396          BRW     SCAN_CLASSES                ; Go on to show classes if no errors
                        020B   397
                        020B   398 SF_ERR:
             01DC   31  020B   399          BRW     SHD_ERR                     ; Go log error and return
```

SHO
V04

63

```
                                         020E  SHOW_INPUT_QUAL:
                            58    DD  020E           PUSHL   R8                          ; save R8 so we can use it as scratch
                        58  01    D0  0210           MOVL    #1,R8                       ; init input file counter
                     53  04  AB    D0  0213           MOVL    4(R11),R3                   ; Point to beginning of FAOL ctrl string
                     59  04  AA    D0  0217           MOVL    4(R10),R9                   ; Point to beginning of FAOL parm list
63   00000060'EF   00000058'EF    28  021B           MOVC3   CS_SEG3,CS_SEG3+8,(R3)      ; Move fixed segment into control string
                            66    D5  0227           TSTL    (R6)                        ; Qualifier present?
                            05    12  0229           BNEQ    5$                          ; Branch if yes
               83      4F4E  8F    B0  022B           MOVW    #^A/NO/,(R3)+               ; No -- move NO to control string
                                      0230  5$:
63   0000006C'EF   00000064'EF    28  0230           MOVC3   CS_SEG4,CS_SEG4+8,(R3)      ; Move fixed segment into control string
                     89  04  A6    D0  023C           MOVL    4(R6),(R9)+                 ; Qualifier name descr to FAOL prmlst
                            66    D5  0240           TSTL    (R6)                        ; Qualifier present?
                            02    12  0242           BNEQ    7$                          ; Branch if yes
                            19    11  0244           BRB     10$                         ; No, go straight to SHOW_DOUBLE call
                                      0246  7$:
                     55  42  A7    9A  0246           MOVZBL  MRB$B_INP_FILES(R7),R5      ; Number of input files to R5
                        01  55    D1  024A           CMPL    R5,#1                       ; How many input files?
                            16    14  024D           BGTR    12$                         ; branch if there are > 1 input files
                                      024F  ;
                                      024F  ; If we got here, only one file was specified for /INPUT, so just
                                      024F  ; do a SHOW_DOUBLE for that file and the /INPUT qualifier, and get out.
                                      024F  ;
63   00000079'EF   00000071'EF    28  024F           MOVC3   CS_SEG5,CS_SEG5+8,(R3)      ; Move fixed segment into control string
                     89  00  B6    D0  025B           MOVL    @(R6),(R9)+                 ; Qual value descr to FAOL prmlst
                                      025F  10$:
                                      025F
                                      025F  ;
                                      025F  ; At this point, either 0 (/NOINPUT case) or 1 file was specified for input.
                                      025F  ;
                                      025F
                          0292    30  025F           BSBW    SHOW_DOUBLE                 ; Show the line, double-spaced
                          0045    31  0262           BRW     25$                         ; get out
                                      0265
                                      0265  ;
                                      0265  ; Begin multi-file summary loop
                                      0265  ;
                                      0265  12$:
                            55    DD  0265           PUSHL   R5                          ; Save count of # of input files
                        01  58    D1  0267           CMPL    R8,#1                       ; Are we on file #1?
                            0E    12  026A           BNEQ    15$                         ; Branch if we have passed file #1
                                      026C  ;
                                      026C  ; If we got here, we are doing a multi-file summary and we are processing
                                      026C  ; input file #1.
                                      026C  ;
63   00000079'EF   00000071'EF    28  026C           MOVC3   CS_SEG5,CS_SEG5+8,(R3)      ; Move fixed segment into control string
                            0C    11  0278           BRB     18$                         ; skip past alternate control string
                                      027A  15$:
                                      027A  ;
                                      027A  ; If we got here, this is a multi-file summary
                                      027A  ; and we have processed the first file in the list already,
                                      027A  ; so we use a different FAO control string
                                      027A  ;
63   0000009D'EF   00000095'EF    28  027A           MOVC3   CS_SEG7,CS_SEG7+8,(R3)      ; Move fixed segment into control string
                                      0286  18$:
                          55    8ED0  0286           POPL    R5                          ; get R5 back (pushed to save from MOVC)
                     89  00  B6    D0  0289           MOVL    @(R6),(R9)+                 ; Qual value descr to FAOL prmlst
```

```
        66   05   C0  028D            ADDL2   #IFB$K_SIZE,(R6)        ; move to next IFB
                      0290
                      0290      20$:
        58   55   D1  0290            CMPL    R5,R8                   ; Is this the last input file?
             05   12  0293            BNEQ    23$                     ; branch if not
           025C   30  0295            BSBW    SHOW_DOUBLE             ; Show the last line, double-spaced
             10   11  0298            BRB     25$                     ; and get out
                      029A      23$:
           0253   30  029A            BSBW    SHOW_SINGLE             ; Show the line, single-spaced
     53   04 AB   D0  029D            MOVL    4(R11),R3               ; Point to beginning of FAOL ctrl string
     59   04 AA   D0  02A1            MOVL    4(R10),R9               ; Point to beginning of FAOL parm list
             58   D6  02A5            INCL    R8                      ; on to next input file.
           FFBB   31  02A7            BRW     12$                     ; Loop
                      02AA
                      02AA      25$:
           58 8ED0    02AA            POPL    R8                      ; Restore R8
                  05  02AD            RSB                             ; Return with status in R0
```

```
                                        02AE   SHOW_FILE_QUAL:
                            66   D5     02AE           TSTL    (R6)                        ; Qualifier present?
                            52   13     02B0           BEQL    SHOW_QUAL                   ; Branch if not
                                        02B2
                                        02B2   ;
                                        02B2   ; Use $PARSE to get filespec with defaults applied
                                        02B2   ; and password removed from access control string.  Note that the file
                                        02B2   ; does not exist yet, but $PARSE will still do the work of
                                        02B2   ; applying defaults and removing the password, if necessary
                                        02B2   ;
                       50   01   D0     02B2           MOVL    #1,R0                       ; set up for post-indexing
                         00 B6   90     02B5           MOVB    @(R6),-
                     0000012C'EF        02B8                   SHOW_FAB + FAB$B_FNS        ; plug FAB with filespec length
      00000124'EF    00 B640     D0     02BD           MOVL    @(R6)[R0],-
                                        02C6                   SHOW_FAB + FAB$L_FNA        ; plug FAB with filespec address
                                        02C6           $PARSE  FAB=SHOW_FAB                ; do the parse
                       1C   50   E8     02D3           BLBS    R0,10$                      ; Branch if OK
           00000000'8F      50   D1     02D6           CMPL    R0,#RMS$_SYN                ; Filespec syntax error?
                            01   13     02DD           BEQL    5$                          ; Branch if so
                            05          02DF           RSB                                 ; unknown error, return with status
                                        02E0   ;
                                        02E0   ; file specification syntax error processing: save the bad qualifier name,
                                        02E0   ; to be reported after SHOW display finishes. Note that we can only
                                        02E0   ; report on one syntax error per SHOW.
                                        02E0   ;
                     000002AF'EF   D5   02E0   5$:     TSTL    ERROR_QUAL                  ; has there already been a syntax error?
                            1C   12     02E6           BNEQ    SHOW_QUAL                   ; branch if so (we can only report first err
        000002AF'EF     04 A6    D0     02E8           MOVL    4(R6), -                    ; load address of offending qualifier
                                        02F0                   ERROR_QUAL
                            12   11     02F0           BRB     SHOW_QUAL                   ; display unparsed spec containing error
                                        02F2   10$:
                     00000153'EF   9A   02F2           MOVZBL  SHOW_NAM+NAM$B_ESL,-        ; Plug descriptor with length
                     000002A7'EF        02F8                   SHOW_SPEC_D                 ; passed back from $PARSE
                66   000002A7'EF   DE   02FD           MOVAL   SHOW_SPEC_D,(R6)            ; point to parsed spec descriptor
                                        0304                                               ; and fall through to SHOW_QUAL
                                        0304   SHOW_QUAL:
                                        0304
                    53   04 AB   D0     0304           MOVL    4(R11),R3                   ; Point to beginning of FAOL ctrl string
                    59   04 AA   D0     0308           MOVL    4(R10),R9                   ; Point to beginning of FAOL parm list
                                        030C
63   00000060'EF     00000058'EF   28   030C           MOVC3   CS_SEG3,CS_SEG3+8,(R3)      ; Move fixed segment into control string
                            66   D5     0318           TSTL    (R6)                        ; Qualifier present?
                            05   12     031A           BNEQ    10$                         ; Branch if yes
                 83   4F4E 8F    B0     031C           MOVW    #^A/NO/,(R3)+               ; No -- move NO to control string
                                        0321   10$:
63   0000006C'EF     00000064'EF   28   0321           MOVC3   CS_SEG4,CS_SEG4+8,(R3)      ; Move fixed segment into control string
                    89   04 A6   D0     032D           MOVL    4(R6),(R9)+                 ; Qualifier name descr to FAOL prmlst
                            66   D5     0331           TSTL    (R6)                        ; Qualifier present?
                            0F   13     0333           BEQL    20$                         ; Branch if no
63   00000079'EF     00000071'EF   28   0335           MOVC3   CS_SEG5,CS_SEG5+8,(R3)      ; Move fixed segment into control string
                       89   66   D0     0341           MOVL    (R6),(R9)+                  ; Qual value descr to FAOL prmlst
                                        0344
                                        0344   ;
                                        0344   ; Display a line showing current qualifier
                                        0344   ;
                                        0344
                                        0344   20$:
```

```
        01AD  30 0344   533         BSBW    SHOW_DOUBLE             ; Show the line, double-spaced
                 0347   534
              05 0347   535         RSB                             ; Return with status in R0
```

```
SHO
Sym

$$.
$$.
$$.
$$.
$$.
ALL
ALL
AVE
BUI
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
CDE
```

```
                              0348    SCAN_CLASSES:
                              0348
                              0348    ;
                              0348    ; Scan the MRB$O_CLASSBITS field of the MRB to determine the selected
                              0348    ; classes. Create a vector of bytes containing a class number for each
                              0348    ; selected class. Then show the selected classes by calling SHOW_CLASSES
                              0348    ; with the vector as input.
                              0348    ;
                              0348    ;
                              0348    ; REGISTER USAGE:
                              0348    ;
                              0348    ;       R0,R1 = scratch
                              0348    ;       R2  = bit field starting position for FFS instruction
                              0348    ;       R3  = bit field size for FFS instruction
                              0348    ;       R4  = class number determined by FFS instruction
                              0348    ;       R5  = index into MRB$O_CLASSBITS
                              0348    ;       R6  = pointer to byte vector of class numbers
                              0348    ;       R7  = pointer to MRB (Monitor Request Block)
                              0348    ;       R8  = pointer to Qualifier Descriptors
                              0348    ;       R9  = class counter
                              0348    ;       R10 = address of descriptor for FAOL parameter list
                              0348    ;       R11 = address of descriptor for FAOL control string
                              0348    ;
                              0348
                              0348            ALLOC   MAX_CLASS_NO+1,R0,R6    ; Get a byte vector for class numbers
                              035D                                            ; ... to be shown, and point R6 to it
                     59   D4  035D            CLRL    R9                      ; Init class counter
                              035F
                              035F    ;
                              035F    ; Use FFS instruction to scan the class bits in the MRB, filling
                              035F    ; the above byte vector with a class number for each class found.
                              035F    ; R9 contains the count of classes found.
                              035F    ;
                              035F
                     55   D4  035F            CLRL    R5                      ; Init starting bit position
                              0361    20$:
                53   20   D0  0361            MOVL    #32,R3                  ; Init bit field size
                              0364                                            ; NOTE -- must handle in 32-bit chunks
                52   55   D0  0364            MOVL    R5,R2                   ; Init start position of next chunk
                              0367    30$:
   54    32 A7  53   52   EA  0367            FFS     R2,R3,MRB$O_CLASSBITS(R7),R4 ; Search for next class number
                              036D                                            ; R4 contains class no. if found
                     12   13  036D            BEQL    40$                     ; Branch if none found this chunk
              6649   54   90  036F            MOVB    R4,(R6)[R9]             ; Move class no. into byte vector
                     59   D6  0373            INCL    R9                      ; Count this class
                              0375
                53   52   C0  0375            ADDL2   R2,R3                   ; Compute next starting
           52   54   01   C1  0378            ADDL3   #1,R4,R2                ; ... position and field size
                53   52   C2  037C            SUBL2   R2,R3                   ; ... for this chunk
                     E6   11  037F            BRB     30$                     ; Go search rest of chunk
                              0381    40$:
FFD8 55    20  0000'8F    3D  0381            ACBW    #MAX_CLASS_NO,#32,R5,20$ ; Loop to process next chunk
```

```
                              0389       ;
                              0389       ; At this point, R6 points to a byte vector consisting of the
                              0389       ; class numbers for classes to show. R9 contains the count of
                              0389       ; classes to show.
                              0389       ;
                              0389
                              0389       ;
                              0389       ; Show class-names and their associated display qualifiers.
                              0389       ;
                              0389
                5B      DD    0389               PUSHL   R11                     ; Stack addr of descr for FAOL ctrl str
                5A      DD    038B               PUSHL   R10                     ; Stack addr of descr for FAOL prm list
                56      DD    038D               PUSHL   R6                      ; Stack addr of byte vector of class nos.
                59      DD    038F               PUSHL   R9                      ; Stack count of classes to show
  0000041E'EF   04      FB    0391               CALLS   #4,SHOW_CLASSES         ; Show all classes and their ...
                              0398                                               ; ... associated display qualifiers
             4F 50      E9    0398               BLBC    R0,SHD_ERR              ; Exit if error
                              039B
                              039B       ;
                              039B       ; Print one final blank line for readability
                              039B       ;
                              039B
                              039B               ALLOC   8,R0,R1                 ; Get quadword for dummy descr
                61      D4    03A8               CLRL    (R1)                    ; Make length 0
      04 A1  04 A1      DE    03AA               MOVAL   4(R1),4(R1)             ; ... and point it to itself
                01      DD    03AF               PUSHL   #1                      ; Stack single spacing indicator
                51      DD    03B1               PUSHL   R1                      ; ... and dummy text descriptor
  00000000'GF   02      FB    03B3               CALLS   #2,G^SCR$PUT_LINE       ; Put blank line to the terminal
             2D 50      E9    03BA               BLBC    R0,SHD_ERR              ; Exit if error
                              03BD
                              03BD
                              03BD       ;
                              03BD       ; SHOW subcommand processing complete
                              03BD       ;
                              03BD
                              03BD
                              03BD       ;
                              03BD       ; See if we encountered a filespec syntax error on SHOW_FILE_QUAL parse,
                              03BD       ; and if so, report it.
                              03BD       ;
  50    00000000'8F     D0    03BD               MOVL    #SS$_NORMAL,R0          ; Assume success status
        000002AF'EF     D5    03C4               TSTL    ERROR_QUAL              ; was there a syntax error?
                36      13    03CA               BEQL    SHD_RET                 ; no, return with success
        000002AF'EF     DD    03CC               PUSHL   ERROR_QUAL              ; push address of qualifier descriptor
                7E      D4    03D2               CLRL    -(SP)                   ; no secondary code
        00000000'8F     DD    03D4               PUSHL   #MNR$_FILSYNERR         ; Push syntax error status code
  00000000'EF   03      FB    03DA               CALLS   #3,MON_ERR
  50    00000000'8F     D0    03E1               MOVL    #MNR$_FILSYNERR,R0      ; Get status to caller
                18      11    03E8               BRB     SHD_RET                 ; Go return
                              03EA
                              03EA       SHD_ERR:                                ; Log error and return
                50      DD    03EA               PUSHL   R0                      ; Bad status on stack
                6E      DF    03EC               PUSHAL  (SP)                    ; Stack pointer to bad status
        00000000'8F     DD    03EE               PUSHL   #MNR$_SHOWERR           ; Stack MONITOR failing status code
  00000000'EF   02      FB    03F4               CALLS   #2,MON_ERR              ; Log the error
  50    00000000'8F     D0    03F8               MOVL    #MNR$_SHOWERR,R0        ; Get status to caller
                              0402
```

```
                           0402   647 SHD_RET:                                        ; Return point from SHODEF_CMD routine
                           0402   648
                  01  BB   0402   649          PUSHR   #^M<R0>                        ; Save return status
                           0404   650
    00000000'GF   00  FB   0404   651          CALLS   #0,G^LIB$PUT_BUFFER            ; Output SCRPKG buffer & stop buffering
                           040B   652
                  00  DD   040B   653          PUSHL   #0                             ; Indicate "clear buffer mode"
    00000000'GF   01  FB   040D   654          CALLS   #1,G^LIB$SET_BUFFER            ; ... and tell SCRPKG to clear it
    00000000'GF   00  FB   0414   655          CALLS   #0,G^SCR$STOP_OUTPUT           ; Stop output stream
                           041B   656
                  01  BA   041B   657          POPR    #^M<R0>                        ; Get back SHODEF_CMD return status
                      04   041D   658          RET                                    ; Return with status in R0
```

```
SHO
Sym

RMS
RV_
SA[
SCA
SCR
SCR
SCR
SCR
SF_
SH[
SHC
SHD
SHD
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SS$
STA
STA
SYS
SYS
SYS
TOP
TOP
TOP
TOP
VIE



PSE
---
.
MON
$AE
$S/
```

```
041E          .SBTTL  SHOW_CLASSES - Show all selected classes
041E
041E  ;++
041E  ;
041E  ; FUNCTIONAL DESCRIPTION:
041E  ;
041E  ;       SHOW_CLASSES fills out the FAOL parameter list and
041E  ;       control string with the information required to
041E  ;       display each of the classes followed by the display
041E  ;       qualifier for each. It accepts as input a count of
041E  ;       the number of classes to show and a byte vector
041E  ;       containing a class number for each class.
041E  ;
041E  ; INPUTS:
041E  ;
041E  ;        4(AP) - count of classes to be shown
041E  ;
041E  ;        8(AP) - address of byte vector containing a class number
041E  ;                for each selected class
041E  ;
041E  ;       12(AP) - address of descriptor for FAOL parameter list
041E  ;
041E  ;       16(AP) - address of descriptor for FAOL control string
041E  ;
041E  ;
041E  ; IMPLICIT INPUTS:
041E  ;
041E  ;       CDBHEAD      - table of contiguous CDB's, one for each class.
041E  ;
041E  ;       CLASSTABLE   - table of contiguous quadwords, one for each class.
041E  ;                       Each quadword consists of a pointer to a counted
041E  ;                       ASCII string for the class name followed by a
041E  ;                       longword containing the class number.
041E  ;
041E  ;       PROCD_TABLE  - table of contiguous longword pointers, one for
041E  ;                       each PROCESSES display qualifier. Each pointer
041E  ;                       points to a string descriptor for the qualifier
041E  ;                       name.
041E  ;
041E  ;       STAT_TABLE   - table similar to PROCD_TABLE but instead points
041E  ;                       to statistic qualifiers for standard classes.
041E  ;
041E  ; OUTPUTS:
041E  ;
041E  ;       none
041E  ;
041E  ; IMPLICIT OUTPUTS:
041E  ;
041E  ;       FAOL control string and parameter list updated.
041E  ;
041E  ; ROUTINE VALUE:
041E  ;
041E  ;       R0 = SS$_NORMAL, or called routine error status.
041E  ;
041E  ; SIDE EFFECTS:
041E  ;
041E  ;       none
```

```
        041E   717 ;
        041E   718 ;--
        041E   719
```

**F

```
                                041E
                          0FFC  041E  .ENTRY  SHOW_CLASSES, ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                                0420
            5A    0C AC    D0   0420          MOVL    12(AP),R10               ; Load ptr to descr for FAOL parm list
            5B    10 AC    D0   0424          MOVL    16(AP),R11               ; Load ptr to descr for FAOL ctrl str
                                0428
                  04 AC    D5   0428          TSTL    4(AP)                    ; Test number of classes
                     03    12   042B          BNEQ    10$                      ; Branch if at least one
                   004B    31   042D          BRW     50$                      ; None -- go tell user
                                0430
                                0430  ;
                                0430  ; Display "Classes:" heading
                                0430  ;
                                0430
                                0430  10$:
                     01    DD   0430          PUSHL   #1                       ; Stack "single-spacing" indicator
           000000D3'EF     DF   0432          PUSHAL  CLASS_HDG                ; ... and text descriptor
       00000000'GF   02    FB   0438          CALLS   #2,G^SCR$PUT_LINE        ; Put header to the terminal
                  52 50    E9   043F          BLBC    R0,SHC_RET               ; Exit if error
                                0442
                     57    D4   0442          CLRL    R7                       ; Init byte vector index
                     58    D4   0444          CLRL    R8                       ; Indicate this class at beg of line
                                0446  20$:
                     58    D5   0446          TSTL    R8                       ; Need to start this class on new line?
                     0E    12   0448          BNEQ    30$                      ; Branch if not
                                044A
            59    04 AA    D0   044A          MOVL    4(R10),R9                ; Point to beginning of FAOL parm list
            53    04 AB    D0   044E          MOVL    4(R11),R3                ; Point to beginning of FAOL ctrl string
           83     20       90   0452          MOVB    #^A/ /,(R3)+             ; Start classes in column 2 of screen
           58     03       D0   0455          MOVL    #3,R8                    ; Init "classes per line" count
                                0458  30$:
            50    08 AC    D0   0458          MOVL    8(AP),R0                 ; Get addr of byte vector
            52     6047    9A   045C          MOVZBL  (R0)[R7],R2              ; Get class number for BUILD_FAOL_ARGS
                   0032    30   0460          BSBW    BUILD_FAOL_ARGS          ; Bld FAOL prmlst & ctrstr 4 this class
                                0463                                           ; NOTE -- this rtn destroys REGS R0-R2
                                0463                                           ; ... and R4-R6. Also updates R3 and R9
                     58    D7   0463          DECL    R8                       ; Update "classes left this line"
                     06    12   0465          BNEQ    40$                      ; Branch if at least one left
                                0467
                                0467  ;
                                0467  ; Show a line of 3 classes
                                0467  ;
                                0467
                   0086    30   0467          BSBW    SHOW_SINGLE              ; Show the line, single-spaced
                  27 50    E9   046A          BLBC    R0,SHC_RET               ; Exit if error
                                046D  40$:
         D4 57    04 AC    F2   046D          AOBLSS  4(AP),R7,20$             ; Loop back to do next class
```

```
               58  D5  0472   769         TSTL    R8                      ; Classes remaining to be shown?
               17  13  0474   770         BEQL    SHC_NORM                ; Branch if not
                       0476   771
                       0476   772 ;
                       0476   773 ; Show final class line
                       0476   774 ;
                       0476   775
             0077  30  0476   776         BSBW    SHOW_SINGLE             ; Show the line, single-spaced
               12  11  0479   777         BRB     SHC_NORM                ; ... and return with normal status
                       047B   778
                       047B   779 ;
                       047B   780 ; Show "No classes" heading
                       047B   781 ;
                       047B   782
                       047B   783 50$:
               01  DD  047B   784         PUSHL   #1                      ; Stack "single-spacing" indicator
      000000E3'EF  DF  047D   785         PUSHAL  NO_CLASS_HDG            ; ... and text descriptor
00000000'GF    02  FB  0483   786         CALLS   #2,G^SCR$PUT_LINE       ; Put header to the terminal
            07 50  E9  048A   787         BLBC    R0,SHC_RET              ; Exit if error
                       048D   788
                       048D   789 SHC_NORM:
50   00000000'8F   D0  048D   790         MOVL    #SS$_NORMAL,R0          ; No failing status hit
                       0494   791
                       0494   792 SHC_RET:
                   04  0494   793         RET                             ; Return with status already in R0
```

```
                                    0495  ;
                                    0495  ; BUILD_FAOL_ARGS subroutine.
                                    0495  ;
                                    0495  ; This subroutine annexes FAOL directives to the control
                                    0495  ; string, and parameters to the parameter list for the
                                    0495  ; current class.
                                    0495  ;
                                    0495  ;
                                    0495  ; REGISTER USAGE:
                                    0495  ;
                                    0495  ;       R0,R1    = scratch
                                    0495  ;       R2       = class number of current class (input)
                                    0495  ;       R3       = next available byte in FAOL control string (input)
                                    0495  ;       R4,R5,R6 = scratch
                                    0495  ;       R9       = next available longword in FAOL parameter list (input)
                                    0495  ;
                                    0495  ;       NOTE -- R3 and R9 are updated. R0-R2 and R4-R6 are destroyed.
                                    0495  ;
                                    0495
                                    0495  BUILD_FAOL_ARGS:
                                    0495
                                    0495  ;
                                    0495  ; Move class-name cstring pointer to FAOL parameter list
                                    0495  ;
                                    0495
         50   00000004'EF    DE     0495          MOVAL   CLASSTABLE+4,R0         ; Get addr of table of class quadwords
                 50   6042   7D     049C          MOVQ    (R0)[R2],R0             ; R0 gets class-name cstring ptr
                    89  50   D0     04A0          MOVL    R0,(R9)+                ; Move it to FAOL parm list
                                    04A3
                                    04A3  ;
                                    04A3  ; Obtain CDB address for this class
                                    04A3  ;
                                    04A3
   56     52   00000053 8F   C5     04A3          MULL3   #CDB$K_SIZE,R2,R6       ; Get CDB offset from class number
                                    04AB                                          ; NOTE - this rtn no longer needs class num
       56    00000000'EF46   9E     04AB          MOVAB   CDBHEAD[R6],R6          ; Get CDB address
                                    04B3
                                    04B3  ;
                                    04B3  ; Move appropriate segment of FAO directives into FAO control string
                                    04B3  ; and move address of descriptor for display qualifier to FAOL parm list,
                                    04B3  ; if one exists.
                                    04B3  ;
                                    04B3
              50    44 A6    9A     04B3          MOVZBL  CDB$B_ST_CUR(R6),R0     ; R0 gets display qualifier index
           09 4B A6    04    E0     04B7          BBS     #CDB$V_STD,CDB$L_FLAGS(R6),10$ ; Branch if standard class
         56   00000000'EF    DE     04BC          MOVAL   PROCD_TABLE,R6          ; Get ptr to PROCESSES display qual table
                        07   11     04C3          BRB     20$                     ; ... and go get desired element
                                    04C5  10$:
         56   00000000'EF    DE     04C5          MOVAL   STAT_TABLE,R6           ; Get ptr to statistic qualifier ...
                                    04CC                                          ; ... table for standard classes
                                    04CC  20$:
                      6640   D5     04CC          TSTL    (R6)[R0]                ; Is there a qual defined for this stat?
                        12   13     04CF          BEQL    30$                     ; Branch if no
                 89   6640   D0     04D1          MOVL    (R6)[R0],(R9)+          ; Move descr ptr to FAOL prmlst
63   000000C6'EF  000000BE'EF 28    04D5          MOVC3   CL_SEG2,CL_SEG2+8,(R3)  ; Move fixed segment into ctrl string
                        0C   11     04E1          BRB     40$                     ; Go return
                                    04E3
```

```
                                       04E3     852 30$:
63   000000B5'EF   000000AD'EF   28   04E3     853            MOVC3   CL_SEG1,CL_SEG1+8,(R3)  ; Move fixed segment into control ...
                                       04EF     854                                            ; ... string (no display qualifier)
                                       04EF     855 40$:
                                 05   04EF     856            RSB
```

```
                              04F0  ;
                              04F0  ; SHOW_SINGLE and SHOW_DOUBLE subroutine.
                              04F0  ; The 2 routine names are alternate entry points to show a line with
                              04F0  ; single-spacing and double-spacing, respectively.
                              04F0  ; Upon entry, R10 points to the FAOL parameter list descriptor,
                              04F0  ; R11 points to the FAOL control string descriptor and R3 to the next
                              04F0  ; available byte in the control string. The routine updates the control
                              04F0  ; string descriptor, destroys R1, and returns status in R0.
                              04F0  ;
                              04F0
                              04F0  SHOW_SINGLE:                                     ; Show a line and advance one line
                   01    DD   04F0          PUSHL   #1                               ; Stack "single-space" indicator
                   02    11   04F2          BRB     SHOW_COMM                        ; Join common code
                              04F4
                              04F4  SHOW_DOUBLE:                                     ; Show a line and advance two lines
                   02    DD   04F4          PUSHL   #2                               ; Stack "double-space: indicator
                              04F6
                              04F6  SHOW_COMM:                                       ; Common point for 2 entry points
          50    04 AB    D0   04F6          MOVL    4(R11),R0                        ; Get address of FAOL control string
      6B    53     50    C3   04FA          SUBL3   R0,R3,(R11)                      ; Compute actual length
                04 AA    DD   04FE          PUSHL   4(R10)                           ; Stack addr of FAOL parm list
                   6B    DF   0501          PUSHAL  (R11)                            ; Stack addr of FAOL ctr str descr
 0000050B'EF       03    FB   0503          CALLS   #3,SHOW_A_LINE                   ; Display one line of SHOW output
                         05   050A          RSB                                      ; Return with status in R0
```

```
                                  050B        .SBTTL  SHOW_A_LINE - Put a line of SHOW to terminal
                                  050B
                                  050B  ;++
                                  050B  ;
                                  050B  ; FUNCTIONAL DESCRIPTION:
                                  050B  ;
                                  050B  ;       SHOW_A_LINE sends one display line of SHOW output to the
                                  050B  ;       terminal via the SCRPKG. The line to display is defined
                                  050B  ;       by an $FAOL control string and parameter list, both of
                                  050B  ;       which are input to this routine.
                                  050B  ;
                                  050B  ; INPUTS:
                                  050B  ;
                                  050B  ;        4(AP) - address of descriptor for $FAOL control string.
                                  050B  ;
                                  050B  ;        8(AP) - address of $FAOL parameter list.
                                  050B  ;
                                  050B  ;       12(AP) - number of display lines to advance after showing a line.
                                  050B  ;
                                  050B  ; IMPLICIT INPUTS:
                                  050B  ;
                                  050B  ;       OUTDSC - quadword string descriptor for $FAOL output buffer.
                                  050B  ;
                                  050B  ; OUTPUTS:
                                  050B  ;
                                  050B  ;       none
                                  050B  ;
                                  050B  ; IMPLICIT OUTPUTS:
                                  050B  ;
                                  050B  ;       SHOW line sent to Screen Package.
                                  050B  ;
                                  050B  ; ROUTINE VALUE:
                                  050B  ;
                                  050B  ;       R0 = SS$_NORMAL, or called routine error status.
                                  050B  ;
                                  050B  ; SIDE EFFECTS:
                                  050B  ;
                                  050B  ;       none
                                  050B  ;
                                  050B  ;--
                                  050B
                            0004  050B  .ENTRY  SHOW_A_LINE,^M<R2>
                                  050D
                                  050D        ALLOC   10,R1,R2                    ; Allocate a descriptor & a word
                                  051A        $FAOL_S CTRSTR=@4(AP), OUTLEN=8(R2), - ; Format the SHOW line
                                  051A                OUTBUF=OUTDSC, PRMLST=@8(AP)
                 22 50       E9   0530        BLBC    R0,SAL_RET                  ; Exit if error
           62    08 A2       3C   0533        MOVZWL  8(R2),(R2)                  ; Move actual text len to descr
   04 A2   00000004'EF       D0   0537        MOVL    OUTDSC+4,4(R2)              ; Move addr of text to descr
                 0C AC       DD   053F        PUSHL   12(AP)                      ; Stack spacing indicator
                    62       DF   0542        PUSHAL  (R2)                        ; ... and text descriptor
      00000000'GF    02      FB   0544        CALLS   #2,G^SCR$PUT_LINE           ; Put the SHOW line to the terminal
                                  054B
                 07 50       E9   054B        BLBC    R0,SAL_RET                  ; Exit if error
      50   00000000'8F       D0   054E        MOVL    #SS$_NORMAL,R0              ; No failing status hit
                                  0555
                                  0555  SAL_RET:
```

```
 04  0555   940          RET                               ; Return with status in R0
     0556   941
     0556   942 .END
```

```
                                                                 I 15
SHODEF                                  - MONITOR SHOW DEFAULT Command          16-SEP-1984 02:05:00  VAX/VMS Macro V04-00          Page  27
Symbol table                                                                     5-SEP-1984 02:02:35  [MONTOR.SRC]SHODEF.MAR;1            (20)

$$.TAB                         = 00000148 R     01    CDB$V_DISKAC                   = 00000006
$$.TABEND                      = 000001A8 R     01    CDB$V_DISKVN                   = 00000007
$$.TMP                         = 00000008             CDB$V_EXPLIC                   = 0000000C
$$.TMP1                        = 00000001             CDB$V_FILLER                   = 0000000D
$$.TMP2                        = 000000CF             CDB$V_HOMOG                    = 00000005
ALLCL_INT_DEFAULT                ********   X   03    CDB$V_KUNITS                   = 0000000A
ALL_STAT                       = 00000000             CDB$V_PERCENT                  = 00000000
AVE_STAT                       = 00000002             CDB$V_QFILLER                  = 00000002
BUILD_FAOL_ARGS                  00000495 R     03    CDB$V_STD                      = 00000004
CDB                            = 00000000             CDB$V_SWAPBUF                  = 00000001
CDB$A_BUFFERS                  = 0000002E             CDB$V_SYSCLS                   = 00000008
CDB$A_CDX                      = 00000032             CDB$V_UNIFORM                  = 00000002
CDB$A_CHDHDR                   = 0000004F             CDB$V_WIDE                     = 0000000B
CDB$A_FAOCTR                   = 00000004             CDB$W_BLKLEN                   = 00000020
CDB$A_ITMSTR                   = 0000001C             CDB$W_DISPCTL                  = 00000036
CDB$A_POSTCOLL                 = 00000026             CDB$W_QFLAGS                   = 00000045
CDB$A_PRECOLL                  = 00000022             CDB$W_QFLAGS_CUR               = 00000049
CDB$A_SUMBUF                   = 0000000C             CDB$W_QFLAGS_DEF               = 00000047
CDB$A_TITLE                    = 00000010             CDBHEAD                          ********   X   03
CDB$B_FAOPRELEN                = 00000041             CLASSTABLE                       ********   X   03
CDB$B_FAOSEGLEN                = 00000040             CLASS_HDG                        000000D3 R     01
CDB$B_ST                       = 00000042             CLASS_HDR                      = 00000000
CDB$B_ST_CUR                   = 00000044             CL_SEG1                          000000AD R     01
CDB$B_ST_DEF                   = 00000043             CL_SEG2                          000000BE R     01
CDB$K_SIZE                     = 00000053             CS_SEG1                          00000027 R     01
CDB$L_BUFFERS                  = 0000002A             CS_SEG2                          00000040 R     01
CDB$L_ECOUNT                   = 00000018             CS_SEG3                          00000058 R     01
CDB$L_FAOCTR                   = 00000000             CS_SEG4                          00000064 R     01
CDB$L_FLAGS                    = 0000004B             CS_SEG5                          00000071 R     01
CDB$L_ICOUNT                   = 00000014             CS_SEG6                          0000007F R     01
CDB$L_MIN                      = 00000038             CS_SEG7                          00000095 R     01
CDB$L_RANGE                    = 0000003C             CT_STR                           0000000F R     01
CDB$L_SUMBUF                   = 00000008             CURR_ERRCODE                     ********   X   03
CDB$M_CPU                      = 00000002             CURR_MRBPTR                      ********   X   03
CDB$M_CPU_COMB                 = 00000008             CUR_STAT                       = 00000001
CDB$M_CTPRES                   = 00000001             DEF$A_DISP                     = 0000000C
CDB$M_DISABLE                  = 00000200             DEF$A_REC                      = 00000004
CDB$M_DISKAC                   = 00000040             DEF$A_SUMM                     = 00000014
CDB$M_DISKVN                   = 00000080             DEF$L_DISP                     = 00000008
CDB$M_EXPLIC                   = 00001000             DEF$L_REC                      = 00000000
CDB$M_HOMOG                    = 00000020             DEF$L_SUMM                     = 00000010
CDB$M_KUNITS                   = 00000400             DEF$S_DEF_DESC                 = 00000018
CDB$M_PERCENT                  = 00000001             DEF_DESC                       = 00000000
CDB$M_STD                      = 00000010             ERROR_QUAL                       000002AF R     01
CDB$M_SWAPBUF                  = 00000002             FAB$B_FNS                      = 00000034
CDB$M_SYSCLS                   = 00000100             FAB$C_BID                      = 00000003
CDB$M_UNIFORM                  = 00000004             FAB$C_BLN                      = 00000050
CDB$M_WIDE                     = 00000800             FAB$C_SEQ                      = 00000000
CDB$S_CDB                      = 00000053             FAB$C_VAR                      = 00000002
CDB$S_FILLER                   = 00000013             FAB$L_ALQ                      = 00000010
CDB$S_FLAGS                    = 00000004             FAB$L_FNA                      = 0000002C
CDB$S_QFILLER                  = 0000000E             FAB$L_FOP                      = 00000004
CDB$S_QFLAGS                   = 00000002             FAB$V_CHAN_MODE                = 00000002
CDB$V_CPU                      = 00000001             FAB$V_FILE_MODE                = 00000004
CDB$V_CPU_COMB                 = 00000003             FAB$V_LNM_MODE                 = 00000000
CDB$V_CTPRES                   = 00000000             FAB$V_NAM                      = 00000018
CDB$V_DISABLE                  = 00000009             FAB$W_GBC                      = 00000048
```

```
                                                                   J 15
SHODEF                           - MONITOR SHOW DEFAULT Command            16-SEP-1984 02:05:00  VAX/VMS Macro V04-00          Page  28
Symbol table                                                                5-SEP-1984 02:02:35  [MONTOR.SRC]SHODEF.MAR;1          (20)

FILE_HDR                         = 00000000                 MNR_HOM$L_ELTCT                  = 00000000
FLUSH_INT_DEFAULT                  ********   X    03       MNR_HOM$L_RESERVED               = 00000004
HOM_CLASS_PRE                    = 00000000                 MNR_HOM$S_HOM_CLASS_PRE          = 00000008
ID_STR                             0000001C R      01       MNR_PRO$B_PRI                    = 0000000A
IFB                              = 00000000                 MNR_PRO$K_DSIZE                  = 0000003B
IFB$A_INPUT                      = 00000000                 MNR_PRO$K_FSIZE                  = 00000040
IFB$B_COL_NO                     = 00000004                 MNR_PRO$K_PSIZE                  = 00000008
IFB$K_SIZE                       = 00000005                 MNR_PRO$K_REV0DSIZE              = 00000033
IFB$S_IFB                        = 00000005                 MNR_PRO$K_REV1DSIZE              = 0000003B
INTERVAL_DEFAULT                   ********   X    03       MNR_PRO$L_BIOCNT                 = 0000002F
LIB$PUT_BUFFER                     ********   X    03       MNR_PRO$L_CPUTIM                 = 0000002B
LIB$SET_BUFFER                     ********   X    03       MNR_PRO$L_DIOCNT                 = 00000023
MAX_CLASS_NO                       ********   X    03       MNR_PRO$L_EFWM                   = 00000037
MAX_STAT                         = 00000004                 MNR_PRO$L_EPID                   = 00000033
MIN_STAT                         = 00000003                 MNR_PRO$L_IPID                   = 00000000
MNR$_FILSYNERR                     ********   X    03       MNR_PRO$L_PAGEFLTS               = 00000027
MNR$_SHOWERR                       ********   X    03       MNR_PRO$L_PCTINT                 = 00000004
MNR_CLS$B_TYPE                   = 00000000                 MNR_PRO$L_PCTREC                 = 00000000
MNR_CLS$K_HSIZE                  = 0000000D                 MNR_PRO$L_STS                    = 0000001F
MNR_CLS$Q_STAMP                  = 00000003                 MNR_PRO$L_UIC                    = 00000004
MNR_CLS$S_CLASS_HDR              = 0000000D                 MNR_PRO$O_LNAME                  = 0000000B
MNR_CLS$S_FILLER                 = 0000000F                 MNR_PRO$S_LNAME                  = 00000010
MNR_CLS$S_FLAGS                  = 00000002                 MNR_PRO$S_PROCESS_CLASS          = 0000003B
MNR_CLS$S_STAMP                  = 00000008                 MNR_PRO$S_PRO_CLASS_PRE          = 00000008
MNR_CLS$V_CONT                   = 00000000                 MNR_PRO$W_GPGCNT                 = 0000001B
MNR_CLS$V_FILLER                 = 00000001                 MNR_PRO$W_PPGCNT                 = 0000001D
MNR_CLS$W_FLAGS                  = 00000001                 MNR_PRO$W_STATE                  = 00000008
MNR_CLS$W_RESERVED               = 0000000B                 MNR_SYI$B_MPCPUS                 = 0000000D
MNR_HDR$B_TYPE                   = 00000000                 MNR_SYI$B_TYPE                   = 00000000
MNR_HDR$K_CLASSBITS              = 00000073                 MNR_SYI$K_BALSETMEM              = 0000001E
MNR_HDR$K_MAXCOMLEN              = 0000003C                 MNR_SYI$K_CPUTYPE                = 00000026
MNR_HDR$K_REVLEVELS              = 00000083                 MNR_SYI$K_MPWHILIM               = 00000022
MNR_HDR$K_SIZE                   = 00000103                 MNR_SYI$K_NODENAME               = 0000000E
MNR_HDR$L_FLAGS                  = 00000001                 MNR_SYI$K_SIZE                   = 0000002A
MNR_HDR$L_INTERVAL               = 00000015                 MNR_SYI$L_BALSETMEM              = 0000001E
MNR_HDR$L_RECCT                  = 00000029                 MNR_SYI$L_CPUTYPE                = 00000026
MNR_HDR$O_CLASSBITS              = 00000073                 MNR_SYI$L_MPWHILIM               = 00000022
MNR_HDR$O_REVOCLSBITS            = 00000019                 MNR_SYI$Q_BOOTTIME               = 00000003
MNR_HDR$Q_BEGINNING              = 00000005                 MNR_SYI$S_BOOTTIME               = 00000008
MNR_HDR$Q_ENDING                 = 0000000D                 MNR_SYI$S_FILLER                 = 0000000E
MNR_HDR$S_BEGINNING              = 00000008                 MNR_SYI$S_FLAGS                  = 00000002
MNR_HDR$S_CLASSBITS              = 00000010                 MNR_SYI$S_NODENAME               = 00000010
MNR_HDR$S_COMMENT                = 0000003C                 MNR_SYI$S_SYS_INFO               = 0000002A
MNR_HDR$S_ENDING                 = 00000008                 MNR_SYI$S_TYPE                   = 00000008
MNR_HDR$S_FILE_HDR               = 00000103                 MNR_SYI$T_NODENAME               = 0000000E
MNR_HDR$S_FILLER                 = 00000020                 MNR_SYI$V_CLUSMEM                = 00000000
MNR_HDR$S_FLAGS                  = 00000004                 MNR_SYI$V_FILLER                 = 00000002
MNR_HDR$S_LEVEL                  = 00000008                 MNR_SYI$V_RESERVED1              = 00000001
MNR_HDR$S_REVOCLSBITS            = 0000001C                 MNR_SYI$W_FLAGS                  = 00000001
MNR_HDR$S_REVLEVELS              = 00000080                 MNR_SYI$W_MAXPRCCT               = 0000000B
MNR_HDR$S_TYPE                   = 00000008                 MON_ERR                            ********   X    03
MNR_HDR$T_COMMENT                = 00000035                 MRB                              = 00000000
MNR_HDR$T_LEVEL                  = 0000002D                 MRB$A_COMMENT                    = 0000002C
MNR_HDR$T_REVLEVELS              = 00000083                 MRB$A_DISPLAY                    = 00000020
MNR_HDR$V_FILLER                 = 00000000                 MRB$A_INPUT                      = 0000001C
MNR_HDR$W_COMLEN                 = 00000071                 MRB$A_RECORD                     = 00000024
MNR_HOM$K_PSIZE                  = 00000008                 MRB$A_SUMMARY                    = 00000028
```

```
MRB$B_INP_FILES                 = 00000042                OUTDSC                           ********    X   03
MRB$K_SIZE                      = 00000045                PROCDISPS                      = 00000005
MRB$L_FLUSH                     = 00000014                PROCD_TABLE                      ********    X   03
MRB$L_INTERVAL                  = 00000010                PROCESS_CLASS                  = 00000000
MRB$L_VIEWING_TIME              = 00000018                PRO_CLASS_PRE                  = 00000000
MRB$M_ALL_CLASS                 = 00000400                QUAL$A_ALL                     = 00000064
MRB$M_BY_NODE                   = 00001000                QUAL$A_AVE                     = 00000074
MRB$M_DISPLAY                   = 00000001                QUAL$A_BEG                     = 00000004
MRB$M_DISP_TO_FILE              = 00000020                QUAL$A_BY_NODE                 = 00000054
MRB$M_DIS_CL_REQ                = 00000100                QUAL$A_CLASS                   = 0000005C
MRB$M_INDEFEND                  = 00000010                QUAL$A_COMM                    = 0000004C
MRB$M_INP_CL_REQ                = 00000040                QUAL$A_CPU                     = 000000AC
MRB$M_MFSUM                     = 00000800                QUAL$A_CUR                     = 0000006C
MRB$M_PLAYBACK                  = 00000008                QUAL$A_DISP                    = 00000034
MRB$M_PROC_REQ                  = 00004000                QUAL$A_END                     = 0000000C
MRB$M_RECORD                    = 00000002                QUAL$A_FLUSH                   = 0000001C
MRB$M_REC_CL_REQ                = 00000080                QUAL$A_INP                     = 0000002C
MRB$M_SUMMARY                   = 00000004                QUAL$A_INT                     = 00000014
MRB$M_SUM_CL_REQ                = 00000200                QUAL$A_ITEM                    = 000000BC
MRB$M_SYSCLS                    = 00002000                QUAL$A_MAX                     = 00000084
MRB$O_CLASSBITS                 = 00000032                QUAL$A_MIN                     = 0000007C
MRB$Q_BEGINNING                 = 00000000                QUAL$A_PCENT                   = 000000B4
MRB$Q_ENDING                    = 00000008                QUAL$A_REC                     = 0000003C
MRB$S_BEGINNING                 = 00000008                QUAL$A_SUMM                    = 00000044
MRB$S_CLASSBITS                 = 00000010                QUAL$A_TOPB                    = 0000009C
MRB$S_ENDING                    = 00000008                QUAL$A_TOPC                    = 0000008C
MRB$S_FLAGS                     = 00000002                QUAL$A_TOPD                    = 00000094
MRB$S_MRB                       = 00000045                QUAL$A_TOPF                    = 000000A4
MRB$V_ALL_CLASS                 = 0000000A                QUAL$A_VIEW                    = 00000024
MRB$V_BY_NODE                   = 0000000C                QUAL$L_ALL                     = 00000060
MRB$V_DISPLAY                   = 00000000                QUAL$L_AVE                     = 00000070
MRB$V_DISP_TO_FILE              = 00000005                QUAL$L_BEG                     = 00000000
MRB$V_DIS_CL_REQ                = 00000008                QUAL$L_BY_NODE                 = 00000050
MRB$V_FILLER                    = 0000000F                QUAL$L_CLASS                   = 00000058
MRB$V_INDEFEND                  = 00000004                QUAL$L_COMM                    = 00000048
MRB$V_INP_CL_REQ                = 00000006                QUAL$L_CPU                     = 000000A8
MRB$V_MFSUM                     = 0000000B                QUAL$L_CUR                     = 00000068
MRB$V_PLAYBACK                  = 00000003                QUAL$L_DISP                    = 00000030
MRB$V_PROC_REQ                  = 0000000E                QUAL$L_END                     = 00000008
MRB$V_RECORD                    = 00000001                QUAL$L_FLUSH                   = 00000018
MRB$V_REC_CL_REQ                = 00000007                QUAL$L_INP                     = 00000028
MRB$V_SUMMARY                   = 00000002                QUAL$L_INT                     = 00000010
MRB$V_SUM_CL_REQ                = 00000009                QUAL$L_ITEM                    = 000000B8
MRB$V_SYSCLS                    = 0000000D                QUAL$L_MAX                     = 00000080
MRB$W_CLASSCT                   = 00000030                QUAL$L_MIN                     = 00000078
MRB$W_FLAGS                     = 00000043                QUAL$L_PCENT                   = 000000B0
NAM$B_ESL                       = 0000000B                QUAL$L_REC                     = 00000038
NAM$B_ESS                       = 0000000A                QUAL$L_SUMM                    = 00000040
NAM$B_NOP                       = 00000008                QUAL$L_TOPB                    = 00000098
NAM$B_RSS                       = 00000002                QUAL$L_TOPC                    = 00000088
NAM$C_BID                       = 00000002                QUAL$L_TOPD                    = 00000090
NAM$C_BLN                       = 00000060                QUAL$L_TOPF                    = 000000A0
NAM$C_MAXRSS                    = 000000FF                QUAL$L_VIEW                    = 00000020
NAM$L_ESA                       = 0000000C                QUAL$S_QUALIFIER_DESC          = 000000C0
NAM$L_RSA                       = 00000004                QUALIFIER_DESC                 = 00000000
NAM$V_SYNCHK                    = 00000003                QUALPTR                          ********    X   03
NO_CLASS_HDG                      000000E3 R     01       REG_PROC                       = 00000000
```

```
                                                                L 15
SHODEF                               - MONITOR SHOW DEFAULT Command          16-SEP-1984 02:05:00  VAX/VMS Macro V04-00         Page  30
Symbol table                                                                  5-SEP-1984 02:02:35  [MONTOR.SRC]SHODEF.MAR;1          (20)

RMS$_SYN                             ********   X   03
RV_STR                               00000000 R     01
SAC_RET                              00000555 R     03
SCAN_CLASSES                         00000348 R     03
SCR$PUT_LINE                         ********   X   03
SCR$SET_OUTPUT                       ********   X   03
SCR$STOP_OUTPUT                      ********   X   03
SCRDSC                               ********   X   03
SF_ERR                               0000020B R     03
SHC_NORM                             0000048D R     03
SHC_RET                              00000494 R     03
SHD_ERR                              000003EA R     03
SHD_RET                              00000402 R     03
SHODEF_CMD                           00000000 RG    03
SHOW_A_LINE                          0000050B RG    03
SHOW_CLASSES                         0000041E RG    03
SHOW_COMM                            000004F6 R     03
SHOW_DOUBLE                          000004F4 R     03
SHOW_FAB                             000000F8 R     01
SHOW_FILESPEC                        000001A8 R     01
SHOW_FILE_QUAL                       000002AE R     03
SHOW_INPUT_QUAL                      0000020E R     03
SHOW_NAM                             00000148 R     01
SHOW_QUAL                            00000304 R     03
SHOW_SINGLE                          000004F0 R     03
SHOW_SPEC_D                          000002A7 R     01
SHO_END                              000000FB R     03
SHO_FILES                            000001B0 R     03
SHO_FLUSH                            00000178 R     03
SHO_INT                              000000A2 R     03
SHO_VIEW                             0000013E R     03
SS$_NORMAL                           ********   X   03
STATS                              = 00000005
STAT_TABLE                           ********   X   03
SYS$FAOL                             ********  GX   03
SYS$PARSE                            ********  GX   03
SYS_INFO                           = 00000000
TOPB_PROC                          = 00000003
TOPC_PROC                          = 00000001
TOPD_PROC                          = 00000002
TOPF_PROC                          = 00000004
VIEWING_DEFAULT                      ********   X   03

                                                +----------------+
                                                ! Psect synopsis !
                                                +----------------+

PSECT name                           Allocation           PSECT No.  Attributes
----------                           ----------           ---------  ----------
.  ABS  .                            00000000  (    0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
MONDATA                              000002B3  (  691.)  01 (  1.)  NOPIC   USR   CON   REL   LCL NOSHR NOEXE   RD     WRT NOVEC QUAD
$ABS$                                00000000  (    0.)  02 (  2.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
$$MONCODE                            00000556  ( 1366.)  03 (  3.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC BYTE
```

```
                                            +-------------------------+
                                            ! Performance indicators !
                                            +-------------------------+

Phase                       Page faults   CPU Time       Elapsed Time
-----                       -----------   --------       ------------
Initialization                       29   00:00:00.09    00:00:00.58
Command processing                  130   00:00:00.79    00:00:03.42
Pass 1                              303   00:00:08.53    00:00:26.39
Symbol table sort                     0   00:00:01.16    00:00:02.40
Pass 2                              178   00:00:02.58    00:00:09.67
Symbol table output                  45   00:00:00.30    00:00:01.48
Psect synopsis output                 2   00:00:00.03    00:00:00.03
Cross-reference output                0   00:00:00.00    00:00:00.00
Assembler run totals                689   00:00:13.48    00:00:43.97

The working set limit was 1650 pages.
48455 bytes (95 pages) of virtual memory were used to buffer the intermediate code.
There were 50 pages of symbol table space allocated to hold 828 non-local and 40 local symbols.
942 source lines were read in Pass 1, producing 28 object records in Pass 2.
37 pages of virtual memory were used to define 22 macros.

                                           +----------------------------+
                                           ! Macro library statistics !
                                           +----------------------------+

Macro library name                         Macros defined
------------------                         --------------
_$255$DUA28:[MONTOR.OBJ]MONLIB.MLB;1                    4
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                          0
_$255$DUA28:[SYSLIB]STARLET.MLB;2                      14
TOTALS (all libraries)                                 18

1007 GETS were required to define 18 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:SHODEF/OBJ=OBJ$:SHODEF MSRC$:SHODEF/UPDATE=(ENH$:SHODEF)+EXECML$/LIB+LIB$:MONLIB/LIB
```

0242 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

MONMSG
LIS

REQUEST
LIS

SHODEF
LIS

MONSUB
LIS

PREPOST
LIS

SUMMBUFF
LIS